

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Высшая математика»

# ПРОГРАММИРОВАНИЕ

*Методические рекомендации к лабораторным работам  
для студентов направления подготовки  
01.03.04 «Прикладная математика»  
очной формы обучения*

Часть 1



Могилев 2021

УДК 004.4  
ББК 32.973  
П78

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Высшая математика» «21» января 2021 г.,  
протокол № 5

Составители: ст. преподаватель А. Н. Бондарев;  
доц. И. И. Маковецкий;  
доц. Д. В. Роголев

Рецензент канд. техн. наук, доц. В. М. Ковальчук

Методические рекомендации разработаны на основе рабочей программы по дисциплине «Программирование» для студентов направления подготовки 01.03.04 «Прикладная математика» и предназначены для использования при проведении лабораторных работ в первом семестре.

Учебно-методическое издание

## ПРОГРАММИРОВАНИЕ

Ответственный за выпуск	В. Г. Замураев
Корректор	И. В. Голубцова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.  
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 56 экз. Заказ №

Издатель и полиграфическое исполнение:  
Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет».  
Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий  
№ 1/156 от 07.03.2019.  
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский  
университет, 2021

## Содержание

Введение.....	4
Лабораторная работа № 1. Алгоритмы.....	5
Лабораторная работа № 2. Введение в Python.....	11
Лабораторная работа № 3. Операции над целочисленными данными.....	14
Лабораторная работа № 4. Операции над вещественными данными.....	17
Лабораторная работа № 5. Линейные алгоритмы.....	20
Лабораторная работа № 6. Оператор ветвления if.....	21
Лабораторная работа № 7. Оператор цикла for.....	24
Лабораторная работа № 8. Оператор цикла while.....	27
Лабораторная работа № 9. Работа со встроенными функциями.....	30
Лабораторная работа № 10. Работа со строками.....	33
Список литературы.....	35

## Введение

Целью изучения дисциплины «Программирование» является формирование у студентов базовых знаний в программировании, развитие навыков постановки, формализации и решения задачи на языке программирования высокого уровня.

В первом семестре, в соответствии с рабочей программой дисциплины, студенты выполняют 10 лабораторных работ, варианты и условия которых приведены в методических рекомендациях.

К защите каждой лабораторной работы студент должен подготавливает отчёт, который оформляется с использованием текстовых редакторов и включает: название и цель работы, формулировку задачи для своего варианта, алгоритм решения, текст программы и результаты ее тестирования.

Программы пишутся на языке Python. Оценивание осуществляется автоматически отправкой исполняемого кода в качестве ответа на тестовое задание.

При подготовке к защите студенту рекомендуется изучить теоретический материал и ответить на контрольные вопросы.

## Лабораторная работа № 1. Алгоритмы

**Цель работы:** приобрести практические навыки разработки блок-схем линейных, циклических, разветвляющихся алгоритмов.

### *1.1 Теоретическая часть*

Алгоритмом называется точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

Алгоритм вычислительного процесса – это алгоритм, объектами, к которым он применяется, являются данные.

Алгоритм решения вычислительной задачи – это совокупность правил преобразования исходных данных в результатные.

Свойства алгоритмов.

1 Дискретность – процесс решения задачи должен быть разбит на последовательность отдельных шагов-команд, которые выполняются одна за другой. Только по завершении одной команды возможно выполнение следующей.

2 Понятность – алгоритм должен содержать только те команды, которые известны исполнителю.

3 Детерминированность – каждый шаг и переход к следующему шагу должны быть точно определены, чтобы его мог выполнить другой человек или устройство, у исполнителя нет возможности принять самостоятельное решение.

4 Конечность – предполагается, что алгоритм заканчивает работу за конечное число шагов.

5 Массовость – алгоритм должен решать не одну задачу, а класс задач.

Выделяют три основных вида вычислительных процессов: линейный, ветвящийся и циклический.

Линейным называется такой вычислительный процесс, при котором все этапы решения задачи выполняются в естественном порядке следования записи этих этапов.

Ветвящимся называется такой вычислительный процесс, в котором выбор направления обработки информации зависит от исходных или промежуточных данных (от результатов проверки выполнения какого-либо логического условия).

Циклом называется многократно повторяемый участок вычислений.

Циклическим называется вычислительный процесс, содержащий один или несколько циклов. По количеству выполнений циклы делятся на циклы с определенным числом повторений и циклы с неопределенным числом повторений. Количество повторений последних зависит от соблюдения некоторого условия, задающего необходимость выполнения цикла. При этом условие может проверяться как в начале цикла, так и в его конце (цикл с предусловием и цикл с постусловием).

Для формализации алгоритмов применяются следующие способы: словесный, формульно-словесный, графический, язык операторных схем, алгоритмический язык. Наибольшей наглядностью обладает графический (блок-схемный)

способ записи алгоритмов.

Блок-схемой называется графическое изображение логической структуры алгоритма, в котором каждый этап процесса обработки информации представляется в виде геометрических символов (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций. Перечень символов, их наименование, отображаемые ими функции, форма и размеры определяются ГОСТ 19.701–90 ЕСПД. *Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения.* Основные виды алгоритмических блоков приведены в таблице 1.

Таблица 1 – Виды и значения алгоритмических блоков

Но-мер	Изображение	Значение
1		Терминатор – отображает начало или конец блок-схемы
2		Данные – символ отображает данные
3		Процесс – символ отображает выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации
4		Решение – символ отображает функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа. Соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути
5		Предопределенный процесс – символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (подпрограмме, модуле)
6		Подготовка – символ отображает модификацию команды или группу команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра); символ циклического повторения части алгоритма
7		Соединитель – символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение
8		Комментарий – символ используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний. Пунктирные линии в символе связаны с соответствующим символом или могут обходить группу символов. Текст комментариев должен быть помещен около ограничивающей структуры
9		Линия – символ отображает поток данных или управления

**Пример** – Привести формульно-словесный алгоритм и построить блок-схему алгоритма вычисления факториала натурального числа.

Согласно определению факториал натурального числа

$$n! = \begin{cases} 1, & n = 0, \\ 1 \cdot 2 \cdot \dots \cdot n, & n \neq 0. \end{cases}$$

Опишем алгоритм в формульно-словесной форме.

1 Описание переменных:

$n$  – натуральное число, факториал которого необходимо найти;

$fact$  – переменная для хранения значения факториала;

$i$  – счетчик цикла.

2 Для решения на первом этапе вводится значение числа  $n$ .

3 Задаются начальные значения переменных  $fact = 1$ .

4 Выполняется проверка условия  $i = 0$ . Если условие верно – переход к выводу значения  $fact$ , иначе – переход к следующему шагу.

5 Организуется цикл по переменной  $i$  от 1 до  $n$ .

6 Для каждого значения  $i$  выполняется тело цикла: переменной  $fact$  присваивается значение  $fact \cdot i$ .

7 После выполнения цикла выводится значение  $fact$ .

Блок-схема решения задачи представлена на рисунке 1.

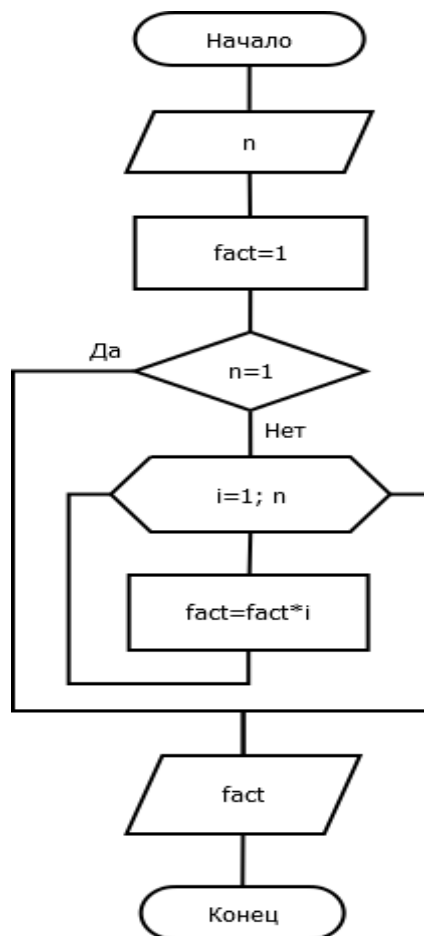


Рисунок 1 – Блок-схема алгоритма для примера

## 1.2 Контрольные вопросы

- 1 Каковы способы записи алгоритмов?
- 2 Кто и когда впервые ввел понятие алгоритма?
- 3 В чем заключаются основные свойства алгоритма?
- 4 Перечислите основные алгоритмические структуры и опишите их.
- 5 Каковы основные принципы разработки алгоритмов?
- 6 Чем объясняется разнообразие форм записи алгоритмов?
- 7 Охарактеризуйте словесно-пошаговый способ записи алгоритмов.
- 8 Охарактеризуйте табличную форму записи алгоритмов.
- 9 Что такое результат выполнения алгоритма?
- 10 Что такое исходные данные?
- 11 Что представляет собой графическая форма записи алгоритма?
- 12 Каков порядок составления блок-схем?
- 13 Охарактеризуйте основные блоки блок-схем.
- 14 Для чего необходимо ветвление в алгоритмах?
- 15 Какие формы ветвления различают?
- 16 Для чего используют структуру «цикл»?
- 17 Какие виды циклов вы знаете?
- 18 Что такое тело цикла?
- 19 Какие циклы называют итерационными? Приведите примеры.
- 20 Что такое итерация?

## 1.3 Задания для самостоятельного выполнения

Постройте алгоритм решения задачи, используя формульно-словесную форму, а также блок-схему решения.

1 Вычислите значение функции. В случае, если значение невозможно вычислить, выведите сообщение «невозможно найти значение».

$$f(x) = \begin{cases} \frac{\ln(x^2 + (2a - 2)x + (a^2 - 2a - 1))}{x + a + 3}, & x \leq 1 - a, \\ \frac{1}{\sqrt{x^3 - a^2x}}, & x > 1 - a. \end{cases}$$

Значения параметра  $a$  совпадают с номером по списку в журнале.

2 Вычислите значение суммы или произведения членов ряда. В случае, если значение невозможно вычислить, выведите сообщение «невозможно вычислить». Значение  $x$  является задаваемым. Варианты задания в соответствии с номером в журнале представлены в таблице 2.



Таблица 2 – Варианты задания 2

Вариант	Выражение	Вариант	Выражение
1	$P = \prod_{k=2}^8 \frac{x \cdot \ln x}{2^k x}$	16	$S = \sum_{k=2}^8 \frac{x \cdot \ln x}{2^k x}$
2	$S = \sum_{n=1}^6 \frac{\sqrt{x} \cdot n}{(n^2 - 2)}$	17	$P = \prod_{n=1}^6 \frac{\sqrt{x} \cdot n}{(n^2 - 2)}$
3	$P = \prod_{n=3}^9 \frac{x \sin nx}{2n + 1}$	18	$S = \sum_{n=2}^5 \frac{x \sin nx}{2n + 1}$
4	$S = \sum_{n=2}^6 \frac{n^2 + \lg nx}{\sqrt{n}}$	19	$P = \prod_{n=2}^6 \frac{n^2 + \lg nx}{\sqrt{n}}$
5	$P = \prod_{n=2}^8 \frac{n + \cos nx}{e^{nx}}$	20	$S = \sum_{n=1}^6 \frac{n + \cos nx}{e^{nx}}$
6	$S = \sum_{n=1}^5 \frac{\sqrt{n} + 2 \sin(nx - 1)}{n^2}$	21	$P = \prod_{n=1}^6 \frac{\sqrt{n} + 2 \sin(nx - 1)}{n^2}$
7	$P = \prod_{n=2}^8 \frac{x \sin nx}{\sqrt{n^2 + 1}}$	22	$S = \sum_{n=2}^8 \frac{x \sin nx}{\sqrt{n^2 + 1}}$
8	$P = \prod_{n=2}^7 \frac{n^2 - 1 + \ln x}{\sqrt{x - 2}}$	23	$S = \sum_{n=2}^7 \frac{n^2 - 1 + \ln x}{\sqrt{x - 2}}$
9	$S = \sum_{n=2}^8 \frac{\sqrt{x} \cdot \sin nx}{n + 10}$	24	$P = \prod_{n=2}^6 \frac{\sqrt{x} \cdot \sin nx}{n + 10}$
10	$P = \prod_{n=1}^9 \frac{1 + n \cos 3x}{2n^2}$	25	$S = \sum_{n=2}^5 \frac{1 + n \cos 3x}{2n^2}$
11	$S = \sum_{n=1}^{10} \frac{x^2}{n + 3 + x}$	26	$P = \prod_{n=1}^{10} \frac{x^2}{n + 3 + x}$
12	$P = \prod_{n=2}^6 \frac{n + x \sin nx}{n^2 \sqrt{x}}$	27	$S = \sum_{n=2}^6 \frac{n + x \sin nx}{n^2 \sqrt{x}}$
13	$S = \sum_{n=1}^9 \frac{n^2 + \cos \sqrt{nx}}{n + 2}$	28	$P = \prod_{n=1}^9 \frac{n^2 + \cos \sqrt{nx}}{n + 2}$
14	$P = \prod_{n=2}^6 \frac{n \ln nx}{\sqrt{n^2 + 5}}$	29	$S = \sum_{n=2}^6 \frac{n \ln nx}{\sqrt{n^2 + 5}}$
15	$S = \sum_{n=2}^8 \frac{1}{nx} \cdot \sin \frac{1}{nx}$	30	$P = \prod_{n=2}^8 \frac{1}{nx} \cdot \sin \frac{1}{nx}$

3 Вычислите значение суммы или произведения. В случае, если это невозможно, выведите сообщение «невозможно вычислить». Значение  $x$  является задаваемым. Варианты заданий представлены в таблице 3.

Таблица 3 – Варианты задания 3

Вариант	Выражение	Вариант	Выражение
1	$P = \prod_{k=2}^8 \frac{(-1)^k x^{2k+1}}{(2k+1)!}$	16	$S = \sum_{k=1}^5 \frac{(-1)^k x^{2k+1}}{(2k+1)!}$
2	$P = \prod_{k=2}^8 \frac{(-1)^k x^{2k-1}}{(2k-1)!}$	17	$S = \sum_{k=1}^5 \frac{(-1)^k x^{2k-1}}{(2k-1)!}$
3	$S = \sum_{k=2}^5 \frac{(-1)^k x^{2k}}{\prod_{i=1}^{k+1} \frac{i}{k}}$	18	$P = \prod_{k=2}^8 \frac{(-1)^k x^{2k}}{\sum_{i=1}^k \frac{x}{i}}$
4	$P = \prod_{k=2}^8 \frac{(-1)^k x^{2k+1}}{\sum_{i=1}^k \frac{\sqrt{x}}{i^2}}$	19	$S = \sum_{k=2}^5 \frac{(-1)^k x^k}{\prod_{i=1}^{k+1} \frac{i}{k^2}}$
5	$S = \sum_{k=2}^5 \frac{(-1)^k x^{2k}}{\prod_{i=1}^{k+1} (2i+1)x}$	20	$P = \prod_{k=2}^8 \frac{(-1)^k x^{2k-1}}{\sum_{i=1}^k \frac{i\sqrt{x}}{k}}$
6	$P = \prod_{k=2}^8 \frac{(-1)^k x^{k-1}}{\sum_{i=1}^k (x-i)}$	21	$S = \sum_{k=2}^5 \frac{(-1)^k x^{2k+1}}{\prod_{i=1}^{k+1} \frac{i}{kx}}$
7	$S = \sum_{k=2}^5 \frac{(-1)^k x^{2k+1}}{\prod_{i=1}^{k+1} i^2}$	22	$P = \prod_{k=2}^8 \frac{(-1)^k x^{2k+1}}{\sum_{i=1}^k \frac{1}{i-x}}$
8	$P = \prod_{k=2}^8 \frac{(-1)^k x^{2k}}{\sum_{i=1}^k \sqrt{i}}$	23	$S = \sum_{k=2}^5 \frac{(-1)^k x^{2k+1}}{\prod_{i=1}^{k+1} \frac{1}{i+1}}$
9	$S = \sum_{n=2}^8 \frac{x^n \cdot \sin nx}{(n+1)!}$	24	$\prod_{n=2}^6 \frac{x^{2n} \cdot \sin nx}{(n+1)!}$
10	$\prod_{n=1}^9 \frac{1+n! \cos 3x}{2n^2}$	25	$\sum_{n=2}^5 \frac{1+n! \cos 3x}{2n^2}$
11	$\sum_{n=1}^{10} \frac{x^{2n+1}}{(n+3)!+x}$	26	$\prod_{n=1}^{10} \frac{x^{2n+1}}{(n+3)!+x}$
12	$\prod_{n=2}^6 \frac{n!+x^{2n-1} \sin nx}{n^2 \sqrt{x}}$	27	$\sum_{n=2}^6 \frac{n!+x^{2n} \sin nx}{n^2 \sqrt{x}}$
13	$\sum_{n=1}^9 \frac{n!+ \cos \sqrt{nx}}{x^{2n}+2}$	28	$\prod_{n=1}^9 \frac{n!+ \cos \sqrt{nx}}{x^{2n}+2}$
14	$\prod_{n=2}^6 \frac{x^{2n} \ln nx}{(n+2)!}$	29	$\sum_{n=2}^6 \frac{x^{2n} \ln nx}{(n+1)!}$
15	$\sum_{n=2}^8 \frac{1}{n!x} \cdot \sin \frac{1}{nx^{2n}}$	30	$\prod_{n=2}^8 \frac{1}{n!x} \cdot \sin \frac{1}{nx^{2n}}$

## Лабораторная работа № 2. Введение в Python

**Цель работы:** ознакомиться со средой разработки Python, изучить основные типы данных, операции ввода и вывода.

### 2.1 Теоретическая часть

Python – объектно-ориентированный, интерпретируемый, переносимый язык сверхвысокого уровня. Программирование на Python позволяет получать быстро и качественно необходимые программные модули. Для разработки на Python можно использовать интегрируемую среду разработки IDLE, которая входит в исходный пакет языка, однако существуют и другие варианты сред для разработки в Python.

Интерпретатор Python может работать в двух режимах: интерактивном и скриптовом. Для запуска интерактивного режима необходимо в командной строке набрать слово Python. Для запуска интерпретатора в скриптовом режиме в командной строке необходимо набрать «python имя\_файла.py».

Идентификаторы в Python – это имена, используемые для обозначения переменных, функций, классов, модулей или других объектов. В идентификаторах допускается использование букв от *a* до *Z*, символ подчеркивания, числа от 0 до 9. Начинаться идентификатор может либо с произвольной буквы, либо с символа подчеркивания. Недопустимо использование знаков препинания или специальных символов в качестве идентификаторов. Python чувствителен к регистру. Существует договоренность для названия идентификаторов: имена классов начинаются с большой буквы, все остальные идентификаторы – с маленькой; знак подчеркивания в начале идентификатора означает, что данный идентификатор закрыт для использования вне класса; если идентификатор начинается и заканчивается двумя знаками подчеркивания, это означает, что он является специальным именем, определенным внутри языка.

Зарезервированными (ключевыми) словами в Python являются: `and`, `as`, `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `exec`, `finally`, `for`, `from`, `global`, `if`, `import`, `in`, `is`, `lambda`, `not`, `or`, `pass`, `print`, `raise`, `return`, `try`, `while`, `which`, `yield`. Недопустимо использование этих слов в качестве имени переменной или иного идентификатора. Все ключевые слова состоят только из букв в нижнем регистре.

Для обозначения отдельных блоков кода в Python используются двоеточия и отступы. По договоренности количество пробелов в отступах равняется четырем, при этом отступ всего блока должен быть одинаковым.

Выражения в Python, как правило, заканчиваются новой строкой, однако существует символ переноса строки (`\`), показывающий, что с окончанием строки не заканчивается код. Выражения, находящиеся внутри скобок, не нуждаются в символе переноса строки.

В Python можно использовать кавычки одинарные (`'`), двойные (`"`) или тройные (`'''`) или (`"""`), чтобы обозначить строчный тип данных, при этом строка должна начинаться и заканчиваться одинаковыми кавычками. Тройные кавычки применяются для ограничения строк, занимающих несколько строк кода.

Символ решетки (#) в Python используется для комментирования. Любые символы после этого символа и до конца строки игнорируются интерпретатором.

Точка с запятой позволяет разместить несколько инструкций в одной строке. В языке Python также используются и другие знаки пунктуации в различных целях.

В язык программирования Python встроены следующие типы данных: None (неопределенное значение переменной), Boolean (логический тип, являющийся подклассом целочисленного типа), int (целый тип), float (число с плавающей запятой), complex (комплексные числа), list (список), tuple (кортеж), range (диапазон), str (строка).

Программа на языке Python состоит из блоков, каждый из которых также может состоять из блоков. Каждый блок в соответствии с концепцией Дейкстра может быть последовательностью инструкций (инструкцией), ветвлением, циклом, подпроцедурой (функцией), классом, модулем, пакетом.

Не рекомендуется в программном коде допускать строки длиной более 80 символов, для длинных строк необходимо выполнить перенос строки с помощью символа (\). Также рекомендуется соблюдать PEP8 (руководство по написанию кода на Python): использовать отступ в 4 пробела, оставлять комментарии в отдельной строке, использовать пробелы вокруг операторов и после запятой, но не непосредственно внутри конструкции скобок, не использовать редкие кодировки. Большинство редакторов кода автоматически поддерживают эти правила.

Ввод и вывод данных осуществляется при помощи команды `input(prompt=None):`

```
a = input()
```

Допускается использование строки подсказки:

```
a = input('Введите значение переменной')
```

Команда ввода `input` по умолчанию возвращает строковое значение, поэтому для организации ввода значений других типов необходимо использовать преобразование типов. Например, для ввода целочисленной переменной следует использовать команду

```
a = int(input('Введите целое число'))
```

Вывод данных осуществляется с помощью команды `print:`

```
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False).
```

Здесь `value` – выводимое значение; `sep` – необязательный параметр, задающий разделитель при выводе нескольких значений; `end` – необязательный параметр, задающий символ конца строки; `file` и `flush` – необязательные параметры, определяющие устройство вывода и управление потоком вывода.

Пример: команда `print('сумма 2 + 3 = ', 2+3)` выдаст на экран сообщение «сумма 2 + 3 = 5».

Для форматированного вывода используется метод `format`, с параметрами и примерами использования которого можно ознакомиться с помощью команды `help('FORMATTING')` в интерактивном режиме Python.

## 2.2 Контрольные вопросы

- 1 Каковы правила написания идентификаторов в языке Python?
- 2 Каким образом осуществляется комментирование в Python?
- 3 Как в Python осуществляется выделение блока кода?
- 4 Перечислите типы данных Python.
- 5 Какую команду используют для организации ввода в Python?
- 6 Какую команду используют для организации вывода в Python?
- 7 Для чего применяется форматированный вывод с помощью метода format?

## 2.3 Задания для самостоятельного выполнения

1 Напишите программу на языке Python, которая запрашивает у пользователя его имя, отчество и фамилию, домашний адрес, возраст, рост в сантиметрах и выводит сообщение в следующем формате:

Здравствуйте, <Фамилия> <Имя>! Ваше отчество <Отчество>, рост <рост> см, возраст <возраст> лет.

Использовать метод format.

2 С помощью форматированного вывода (не используя символы табуляции и пробела) напишите программу на языке Python, которая выводит на экран слово с помощью символов “\*”. Например, для вывода на экран буквы

```
*      *
**     **
* * * *
*  *  *
*      *
*      *
```

текст программы будет таким:

```
print (" {0:1}{0:>6} ".format ("*"))
print (" {0:1}{0:>5} ".format ("**"))
print (" {0:1}{0:>2}{0:>2}{0:>2} ".format ("*"))
print (" {0:1}{0:>3}{0:>3} ".format ("*"))
print (" {0:1}{0:>6} ".format ("*"))
print (" {0:1}{0:>6} ".format ("*"))
```

Шаблоны начертания букв размещены на образовательном портале <http://moodle.bru.by> в курсе «Программирование», раздел «Материалы для лабораторных работ».

Варианты заданий для решения представлены в таблице 4.

Таблица 4 – Варианты задания 2

Вариант	Слово	Вариант	Слово	Вариант	Слово
1	DATA	5	DISK	9	COPY
2	BYTE	6	UNIT	10	BURN
3	WORD	7	SAVE	11	BOOT
4	PORT	8	LOAD	12	BACK

## Окончание таблицы 5

Вариант	Слово	Вариант	Слово	Вариант	Слово
13	CARD	19	PLUG	25	LINK
14	SORT	20	EDIT	26	WIRE
15	ROOT	21	FREE	27	OPEN
16	READ	22	FIRM	28	SOFT
17	INIT	23	FARM	29	MAIL
18	TEXT	24	WARE	30	CODE

### Лабораторная работа № 3. Операции над целочисленными данными

**Цель работы:** ознакомиться с целочисленным типом и основными математическими операциями над целыми числами.

#### 3.1 Теоретическая часть

Целые числа в Python имеют тип `int` и записываются как строки, состоящие из десятичных цифр. Целочисленный тип имеет неограниченную точность, числа могут принимать сколь угодно большие значения. Тип `int` является неизменяемым объектом, выполнение операций над целыми числами приводит к созданию нового числового объекта.

Данные типа `int` поддерживают арифметические, побитовые операции и операции сравнения.

В языке Python тип `int` является классом `int()`, который позволяет преобразовывать строку в целое число типа `int` с учетом указанного основания системы счисления (10, 16, 8, 2); преобразовывать вещественные числа в тип `int` отбрасыванием дробной части; преобразовывать двоичные, восьмеричные и шестнадцатеричные литералы в числа типа `int`. Следует помнить, что невозможно преобразование к типу `int` комплексных чисел, а также строк с записью вещественного числа (с точкой).

Для визуального разграничения разрядов в больших числах следует использовать символ подчеркивания (начиная с Python 3.6).

Для записи шестнадцатеричных чисел литерал числа должен начинаться с комбинации символов `0x` или `0X`. Литералы восьмеричных чисел начинаются с комбинации `0o` или `0O`, двоичные – `0b` или `0B`.

Все вышеуказанные литералы создают объекты целых чисел. Для преобразования целого числа в строку с представлением в любой из трех систем счисления можно использовать функции `hex()`, `oct()`, `bin()`.

Арифметические операции над целыми числами поддерживают следующие операции согласно приоритетам выполнения:

- $x + y$  – сумма чисел  $x$  и  $y$ ;
- $x - y$  – разность чисел  $x$  и  $y$ ;
- $x * y$  – произведение чисел  $x$  и  $y$ ;

$x / y$  – результат от деления  $x$  на  $y$ ;  
 $x // y$  – результат от деления  $x$  на  $y$  с округлением до ближайшего целого числа в меньшую сторону. Операция называется целочисленным делением;  
 $x \% y$  – остаток от деления  $x$  на  $y$ ;  
 $-x$  – значение, противоположное  $x$ ;  
 $+x$  – не изменяет значения  $x$ ;  
 $\text{abs}(x)$  – абсолютная величина  $x$ ;  
 $\text{int}(x)$  – число  $x$  преобразуется в целый тип `int`;  
 $\text{float}(x)$  – число  $x$  преобразуется в число с плавающей точкой;  
 $\text{divmod}(x, y)$  – результатом будет пара  $(x // y, x \% y)$ ;  
 $\text{pow}(x, y)$ ,  $x ** y$  – результатом будет число  $x$  в степени  $y$ .  $\text{pow}(0, 0) = 1$ .  
 Побитовые операции:  
 $x \ll y$  – побитовый сдвиг влево;  
 $x \gg y$  – побитовый сдвиг вправо;  
 $x \& y$  – побитовое И;  
 $x \wedge y$  – побитовое исключающее ИЛИ;  
 $x | y$  – побитовое ИЛИ;  
 $\sim x$  – побитовое НЕ.

Битовые операторы имеют более низкий приоритет по сравнению с арифметическими операторами. Положительные числа расширяются влево неограниченной строкой битов 0. Отрицательные числа, для представления которых используется дополнительный код, расширяются влево неограниченной строкой битов 1.

Операции сравнения в Python имеют одинаковый приоритет, который выше, чем у логических операций.

Разрешенные операции сравнения:

$x < y$  – строго  $x$  меньше  $y$ ;  
 $x \leq y$  –  $x$  меньше или равно  $y$ ;  
 $x > y$  – строго  $x$  больше  $y$ ;  
 $x \geq y$  –  $x$  больше или равно  $y$ ;  
 $x == y$  –  $x$  равно  $y$ ;  
 $x != y$  –  $x$  не равно  $y$ .

Сравнения могут быть связаны произвольно и записаны в цепочки сравнений, в которых для соединения сравнений используются неявные логические операторы AND, OR и NOT.

### 3.2 Контрольные вопросы

- 1 Перечислите все литералы, которые преобразуются к целочисленному типу преобразованием `int()`.
- 2 Допустима ли запись целого числа `0o12345678` и почему?
- 3 Какой тип данных генерируют команды `hex()`, `oct()`, `bin()`?
- 4 Объясните математический смысл операций побитового сдвига « $\ll$ » и « $\gg$ ».
- 5 Объясните математический смысл операции «исключающее или» « $\wedge$ ».

6 Укажите порядок выполнения действий в выражении  $2 ** 3 > 2 + 3$  и объясните полученный результат.

### 3.3 Задания для самостоятельного выполнения

1 Напишите программу на языке программирования Python, которая получает с клавиатуры длину ребра куба (целое число) и выводит на экран его объем и площадь поверхности в одной строке через два пробела.

2 Напишите программу на языке программирования Python, которая получает с клавиатуры три целых числа – длину, ширину и высоту прямоугольного параллелепипеда и выводит на экран в одной строке его объем и площадь поверхности через запятую.

3 Напишите программу на языке программирования Python, которая получает с клавиатуры два целых положительных числа и выводит в одной строке через три пробела их среднее арифметическое и среднее геометрическое.

4 Напишите программу на языке программирования Python, которая получает с клавиатуры два целых положительных числа и выводит в одной строке через три пробела среднее арифметическое и среднее геометрическое их модулей.

5 Напишите программу на языке Python, которая получает с клавиатуры пять целых чисел и выводит на экран сумму всех чисел, которые больше, чем  $a$ , где  $a$  – ваш номер по списку + 10. Запрещено использовать условный оператор. Переменную  $a$  задать в программном коде.

6 В одном городе есть только один проспект, на котором автобусные остановки занумерованы целыми положительными числами. По проспекту организовано движение двух автобусов, про которые известны только номера начала и конца маршрута (вводятся с клавиатуры – сначала начало и конец первого маршрута, потом начало и конец второго маршрута). Подсчитайте и выведите на экран количество остановок, на которых пассажир может совершить пересадку с одного маршрута на другой.

Напишите программу на языке Python для решения этой задачи. Запрещено использовать условный оператор. Подсказка: изучите справку о командах `max()`, `min()`.

7 Напишите программу на языке Python, которая получает с клавиатуры целое число и выводит на экран количество единиц, десятков, сотен и тысяч в этом числе.

8 Напишите программу на языке Python, которая присваивает двум переменным целочисленные значения, полученные с клавиатуры, и меняет значения переменных местами (`swap`). Запрещено использовать дополнительную переменную.

9 С клавиатуры вводится число целое  $a$ . Напишите программу на языке программирования Python, которая вычисляет  $a^5$  и  $a^{19}$  за пять операций. Допускается использовать только операцию умножения.

10 Даны три целых числа – длина и ширина прямоугольника, сторона квадрата. Определите количество квадратов, которые можно вырезать из прямоугольника. Напишите программу для решения задачи на языке Python.



11 С клавиатуры вводится трехзначное целое положительное число. Выведите на экран число, полученное при перестановке второй и третьей цифр этого числа.

12 Положение ладьи на шахматном поле задается двумя натуральными числами, не превосходящими 8. Получите с клавиатуры координаты ладьи, а также координаты некоторой фигуры на шахматном поле и выведите True, если ладья угрожает фигуре, False – в противном случае. Запрещено пользоваться условным оператором.

## Лабораторная работа № 4. Операции над вещественными данными

**Цель работы:** ознакомиться с типом данных float и основными математическими операциями над числами с плавающей точкой, библиотекой math.

### 4.1 Теоретическая часть

Числа с плавающей точкой (тип float) представляют собой последовательность десятичных цифр, которая включает десятичную точку (.), суффикс научного представления числа  $e$  или  $E$ , за которым следует необязательный знак + или – и одна или несколько цифр. Литерал вещественного числа не может начинаться с  $e$  или  $E$ , первым символом обязательно должна быть цифра или точка.

Числа с плавающей точкой поддерживают арифметические операции и операции сравнения, аналогичные операциям над целыми числами.

В языке Python тип float представлен классом float(), который позволяет преобразовать строку с записью целого числа в тип float; преобразовать строку с записью вещественного числа в тип float; преобразовать целые числа типа int в тип float; принимает строки 'nan' – не число и 'inf' – положительная или отрицательная бесконечность.

При использовании типа данных float следует принимать во внимание проблемы точности этого типа данных. Это связано с особенностями хранения вещественных чисел и их внутренним представлением.

Некоторые полезные методы класса float():

as\_integer\_ratio() – возвращает пару целых чисел, первое из которых равно числителю, а второе – всегда положительному знаменателю обыкновенной дроби, значение которой точно совпадает с указанным исходным числом типа float;

is\_integer() – возвращает True, если дробная часть числа равна 0, и False, если нет.

Для более эффективной обработки числовых данных в Python существует стандартная библиотека math, подключение которой в программном коде выполняется командой import math, для обращения к методам библиотеки следует ис-

пользовать запись `math.<метод>`. Избежать набора префикса можно, импортировав все методы библиотеки командой `from math import *`.

Список наиболее популярных методов этой библиотеки:

`sqrt(a)` – квадратный корень из  $a$ ;

`log(a)` – натуральный логарифм из  $a$ ;

`fabs(a)` – модуль числа  $a$ ;

`round(a)` – округление до ближайшего целого;

`round(a, b)` – округление числа  $a$  до  $b$  знаков после точки;

`floor(a)` – округление  $a$  до меньшего целого значения;

`ceil(a)` – округление  $a$  до большего целого значения;

`isfinite(a)` – проверяет, является ли  $a$  числом;

`modf(a)` – возвращает целую и дробную части  $a$ ;

`sin(a)` – синус угла  $a$ , указанного в радианах;

`cos(a)` – косинус угла  $a$ , указанного в радианах;

`tan(a)` – тангенс угла  $a$ , указанного в радианах.

## 4.2 Контрольные вопросы

1 Опишите тип данных `float`.

2 Перечислите арифметические операции, поддерживаемые типом данных `float`.

3 Объясните результат выполнения строки кода `0.1 + 0.2 == 0.3`.

4 Каков результат выполнения команды `ceil(-2.65)` и почему?

## 4.3 Задания для самостоятельного выполнения

1 Запишите на языке Python выражения для вычисления значения арифметического выражения  $z = \frac{f(\pi + g(t)) \cdot g(b + f(t))}{h(a + f(b + g(t)))}$  и вычислите его значение при

заданных значениях переменных. Арифметические выражения представлены в таблице 5.

Таблица 5 – Варианты задания 1

Номер варианта	$f(t)$	$g(t)$	$h(t)$	Значения переменных
1	$\arctg t$	$\sqrt[5]{t}$	$\ln t$	$a = 1, b = 1, t = 5$
2	$\sqrt[5]{t}$	$\arctg t$	$\ln t$	$a = 1, b = 1, t = 5$
3	$\arctg t$	$\ln t$	$\sqrt[5]{t}$	$a = 1, b = 1, t = 5$
4	$\sqrt[5]{t}$	$\ln t$	$\arctg t$	$a = 1, b = 1, t = 5$
5	$\ln t$	$\sqrt[5]{t}$	$\arctg t$	$a = 1, b = 1, t = 5$
6	$\ln t$	$\arctg t$	$\sqrt[5]{t}$	$a = 1, b = 1, t = 5$
7	$\sin t$	$\sqrt[3]{t}$	$\arctg t$	$a = 2, b = 1, t = 3$

## Окончание таблицы 5

Номер варианта	$f(t)$	$g(t)$	$h(t)$	Значения переменных
8	$\sin t$	$\operatorname{arctg} t$	$\sqrt[3]{t}$	$a = 2, b = 1, t = 3$
9	$\sqrt[3]{t}$	$\sin t$	$\operatorname{arctg} t$	$a = 2, b = 1, t = 3$
10	$\sqrt[3]{t}$	$\operatorname{arctg} t$	$\sin t$	$a = 2, b = 1, t = 3$
11	$\sin t$	$\operatorname{arctg} t$	$\sqrt[3]{t}$	$a = 2, b = 1, t = 3$
12	$\sin t$	$\sqrt[3]{t}$	$\operatorname{arctg} t$	$a = 2, b = 1, t = 3$
13	$\operatorname{tg} t$	$\ln(t)$	$\sqrt[3]{t}$	$a = 0.2, b = 0.1, t = 1.2$
14	$\sqrt[3]{t}$	$\ln(t)$	$\operatorname{tg} t$	$a = 0.2, b = 0.1, t = 1.2$
15	$\operatorname{tg} t$	$\sqrt[3]{t}$	$\ln(t)$	$a = 0.2, b = 0.1, t = 1.2$
16	$\sqrt[3]{t}$	$\operatorname{tg} t$	$\ln(t)$	$a = 0.2, b = 0.1, t = 1.2$
17	$\ln(t)$	$\sqrt[3]{t}$	$\operatorname{tg} t$	$a = 0.2, b = 0.1, t = 1.2$
18	$\ln(t)$	$\operatorname{tg} t$	$\sqrt[3]{t}$	$a = 0.2, b = 0.1, t = 1.2$
19	$\sqrt[3]{t}$	$\cos t$	$\operatorname{arctg} t$	$a = 0.2, b = 0.1, t = 1.2$
20	$\sqrt[3]{t}$	$\operatorname{arctg} t$	$\cos t$	$a = 0.2, b = 0.1, t = 1.2$
21	$\operatorname{arctg} t$	$\sqrt[3]{t}$	$\cos t$	$a = 0.2, b = 0.1, t = 1.2$
22	$\operatorname{arctg} t$	$\cos t$	$\sqrt[3]{t}$	$a = 0.2, b = 0.1, t = 1.2$
23	$\cos t$	$\sqrt[3]{t}$	$\operatorname{arctg} t$	$a = 0.2, b = 0.1, t = 1.2$
24	$\cos t$	$\operatorname{arctg} t$	$\sqrt[3]{t}$	$a = 0.2, b = 0.1, t = 1.2$
25	$\operatorname{arctg} t$	$\sqrt[5]{t}$	$\ln t$	$a = 1, b = 1, t = 2$
26	$\sqrt[5]{t}$	$\operatorname{arctg} t$	$\ln t$	$a = 1, b = 0.3, t = 2$
27	$\operatorname{arctg} t$	$\ln t$	$\sqrt[5]{t}$	$a = 0.3, b = 1, t = 2$
28	$\sqrt[5]{t}$	$\ln t$	$\operatorname{arctg} t$	$a = 1, b = 0.5, t = 2$
29	$\ln t$	$\sqrt[5]{t}$	$\operatorname{arctg} t$	$a = 0.2, b = 1, t = 2$
30	$\ln t$	$\operatorname{arctg} t$	$\sqrt[5]{t}$	$a = 1, b = 0.6, t = 2$

2 Напишите программу на языке программирования Python для вычисления площади прямоугольного треугольника. Длины катетов являются вещественными числами и вводятся с клавиатуры. Ответ выведите с точностью до двух знаков.

3 Напишите программу на языке программирования Python, которая по введенному значению угла (в градусах) выводит значения синуса, косинуса, тангенса и котангенса для этого угла с точностью до четырех знаков.

4 С клавиатуры вводится четыре числа. Напишите программу на языке программирования Python, которая выводит на экран количество положительных чисел среди введенных. Запрещено использовать условный оператор.

## Лабораторная работа № 5. Линейные алгоритмы

**Цель работы:** освоить технику написания линейных алгоритмов на языке программирования Python.

### 5.1 Теоретическая часть

Линейный алгоритм – это алгоритм, в котором операции выполняются последовательно. Блок-схема такого алгоритма представлена на рисунке 2.



Рисунок 2 – Блок-схема линейного алгоритма

### 5.2 Контрольные вопросы

- 1 Какой алгоритм называется линейным?
- 2 Какой общий вид линейного алгоритма?

### 5.3 Задания для самостоятельного выполнения

1 Напишите программу на языке Python, вычисляющую площадь круга и длину окружности заданного радиуса. Отформатируйте выводимый результат,

чтобы в дробной части выводилось два знака.

2 Три точки  $A$ ,  $B$ ,  $C$  на координатной прямой заданы координатами. Напишите программу на языке программирования Python, которая выводит длины отрезков  $AC$  и  $BC$ .

3 Заданы три числа  $A$ ,  $B$ ,  $C$ . Напишите программу на языке программирования Python, которая выводит наименьшее и наибольшее из этих чисел.

4 Заданы три числа  $A$ ,  $B$ ,  $C$ . Напишите программу на языке программирования Python, которая выводит True, если существует треугольник с такими сторонами, False, если такого треугольника не существует.

5 Напишите программу на языке Python, которая переводит введенное с клавиатуры количество информации в байтах в килобайты и мегабайты.

## Лабораторная работа № 6. Оператор ветвления if

**Цель работы:** научиться формализовывать алгоритмы со структурой ветвления.

### 6.1 Теоретическая часть

Условная инструкция if-elif-else (оператор ветвления) – основной инструмент выбора в Python. Синтаксис:

```
if expr1:
    state1
elif expr2:
    state2
else:
    state3
```

Блок-схема, соответствующая этому программному коду, изображена на рисунке 3.

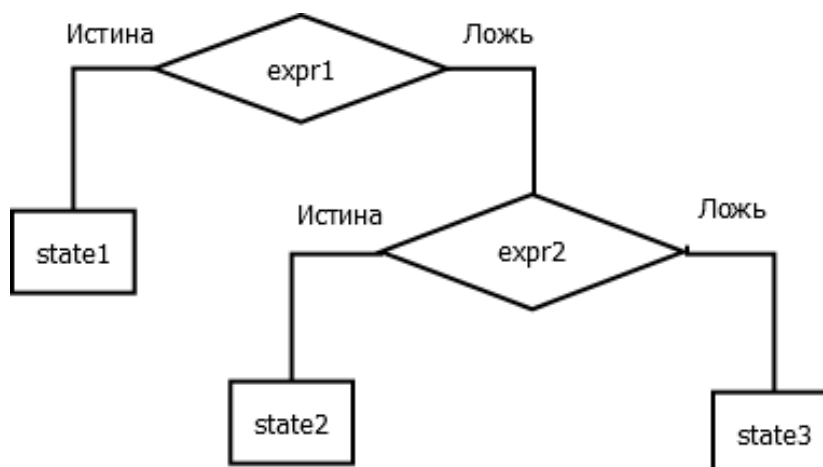


Рисунок 3 – Блок-схема оператора ветвления

Помните, в Python с помощью отступов блоки кода отделяются друг от друга.

При использовании условного оператора следует помнить, что:

- любое число, не равное 0, или непустой объект эквивалентны логическому значению True;
- числа, равные 0, пустые объекты или значение None эквивалентны логическому значению False;
- операции сравнения применяются к структурам рекурсивно;
- операции сравнения возвращают значения True или False;
- неявные логические операторы and, or и not возвращают значения True или False.

В языке Python присутствует тернарный оператор ветвления:

```
A = Y if X else Z
```

В данной инструкции интерпретатор выполнит выражение Y, если X истинно, в противном случае выполнится выражение Z.

## 6.2 Контрольные вопросы

- 1 Что называется разветвляющимся алгоритмом?
- 2 Какой синтаксис у оператора ветвления в языке Python?
- 3 Опишите принцип работы оператора ветвления в языке Python.
- 4 В чём особенность тернарного оператора ветвления?

## 6.3 Задания для самостоятельного выполнения

1 С клавиатуры вводится целое число от 1 до 99. Напишите программу на языке программирования Python, которая выводит это число и добавляет к нему слово «копейка» в правильной форме. Например: 1 копейка, 5 копеек, 32 копейки.

2 С клавиатуры вводится пятизначное число. Напишите программу на языке Python, которая определяет, является ли введенное число палиндромом.

3 С клавиатуры вводятся два числа – первое целое  $d1$  от 1 до 365, второе  $d2$  целое от 1 до 7. Напишите программу на языке Python, которая определяет, какой день недели выпадает на число  $d1$ , если первое января является днем недели с номером  $d2$ .

4 С клавиатуры вводятся два числа – координаты некоторой точки на плоскости. Напишите программу на языке программирования Python, которая определяет, принадлежит точка области, ограниченной параболой  $y = a - x^2$  и прямой  $y = -a$  ( $a$  – номер в журнале).

5 Даны координаты трех вершин параллелограмма. Напишите программу на языке Python, которая вычисляет координаты четвертой вершины.

6 Поезд прибывает на станцию в  $a$  часов  $b$  минут и отправляется в  $c$  часов  $d$  минут. Пассажир пришел на платформу в  $n$  часов  $m$  минут. Напишите программу на языке программирования Python, которая выводит сообщение, стоит ли поезд

на платформе. Числа  $a, b, c, d, n, m$  – целые,  $0 \leq a, c, n \leq 23$ ,  $0 \leq b, d, m \leq 59$ .

7 Дано натуральное число  $n$  ( $1 \leq n \leq 9999$ ), определяющее стоимость товара в копейках. Выразите стоимость в рублях и копейках, например 5 рублей 27 копеек, 2 рубля 31 копейка, 1 рубль ровно и т. п.

8 Для функции, заданной графически (рисунок 4), определите значение  $y$  при заданном значении  $x$ .  $a$  – номер по журналу.

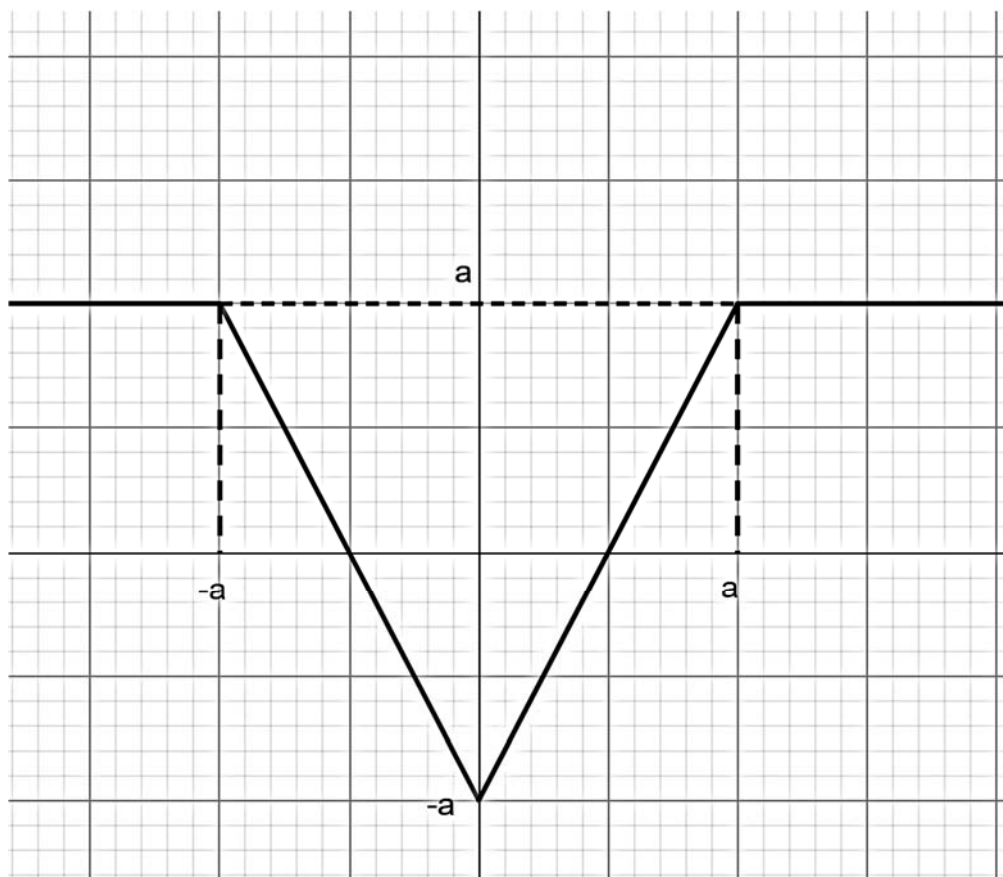


Рисунок 4 – Графическое задание функции

9 Дата некоторого дня определяется тремя натуральными числами:  $g$  (год),  $m$  (порядковый номер месяца) и  $n$  (число). По данным  $g, m$  и  $n$  определите дату предыдущего дня и дату следующего дня. Считайте, что год может быть високосным.

10 Дано целое число  $k$  ( $1 \leq k \leq 180$ ) и последовательность цифр 10111213...9899, в которой выписаны подряд все двузначные числа. Определите  $k$ -ю цифру. Не использовать строковые переменные.

11 Дано целое число  $k$  ( $1 \leq k \leq 253$ ) и последовательность цифр 505152..9899100101...149150, в которой выписаны подряд все натуральные числа от 50 до 150. Определите  $k$ -ю цифру. Не используйте строковые переменные.

12 Дано целое число  $k$  ( $1 \leq k \leq 222$ ) и последовательность цифр 123...91011...9899100101...109110, в которой выписаны подряд все натуральные числа от 1 до 110. Определите  $k$ -ю цифру. Не используйте строковые переменные.

## Лабораторная работа № 7. Оператор цикла `for`

**Цель работы:** научить студентов применению циклических алгоритмов с заданным числом повторений.

### 7.1 Теоретическая часть

Одним из способов запрограммировать циклический алгоритм в языке программирования Python является использование цикла `for`. Цикл `for` в Python обладает способностью перебирать элементы любого типа данных, представляющего собой последовательности (например, строки `str`, кортежи `tuple` или списки `list`) или любые другие итерируемые объекты.

Слово «итерируемые» обозначает, что элементы, содержащиеся внутри такого объекта, перечислимы и расположены в определенном порядке.

Синтаксис оператора `for`:

```
for expr in sequence:
    state1
...
    state2
else:
    state3
...
    state4
```

Простейший способ создания циклической структуры – использование функции `range()`, которая генерирует последовательность чисел, регулирующую количеством переданных в нее аргументов. Синтаксис этой функции:

`range(finish)` – генерирует последовательность целых чисел от 0 до  $finish-1$ ;

`range(start, finish)` – генерирует последовательность целых чисел от  $start$  до  $finish-1$ ;

`range(start, finish, step)` – генерирует последовательность чисел от  $start$  до  $finish-1$  с шагом  $step$ .

При запуске такого цикла в переменную цикла по очереди будут передаваться значения из созданной последовательности, пока не будет достигнут последний ее элемент.

Еще одним способом создания цикла с заданным числом повторений является функция `enumerate()`, которая создает из любых итерируемых объектов кортеж. Кортежи строятся по принципу (индекс элемента, элемент), что бывает крайне удобно, когда, кроме самих элементов, требуется еще и их индекс.

В случае, если требуется параллельно обработать две или более последовательностей, извлекая элементы из каждой из них, следует использовать функцию `zip()`. Функция `zip(s, t)` объединит последовательности  $s$  и  $t$  в последовательность пар  $(s[0], t[0])$ ,  $(s[1], t[1])$  и т. д., остановившись после исчерпания элементов в самой короткой из последовательностей.

Для прерывания цикла используется команда `break`, которая прекращает выполнение цикла и выходит из него.



Команда `continue` прерывает текущую операцию и переходит к следующей, продолжая выполнение цикла.

Если требуется выполнить цикл заданное число раз, при этом значения переменной цикла не важны, можно использовать следующий генератор цикла:

```
for _ in range(*количество_повторений):
    state1
    ...
    state2
```

Разберитесь, как работает данная конструкция, ее использование позволяет сэкономить оперативную память.

Для организации обратного цикла (когда переменная цикла уменьшается) достаточно в функции `range()` в качестве параметра шага указать отрицательное значение.

Особенностью языка программирования Python является наличие так называемых генераторов, или иными словами «циклов в одну строку», позволяющих генерировать списки (`list`). Синтаксис генератора таков:

```
list_name = [x for x in sequence if condition]
```

Например, чтобы создать список всех чисел от 1 до 100, не кратных 3, достаточно написать команду

```
mylist = [x for x in range(1, 101) if x % 3 != 0]
```

## 7.2 Контрольные вопросы

- 1 Опишите принцип работы цикла `for` в языке Python.
- 2 Какая функция используется для создания числовых последовательностей?
- 3 Каким образом можно в Python запустить обратный цикл?
- 4 Можно ли сделать цикл `for` бесконечным и почему?
- 5 Каким образом можно создать список, содержащий только четные числа от 2 до некоторого значения?

## 7.3 Задания для самостоятельного выполнения

1 Напишите программу на языке программирования Python для вычисления суммы из задания 3 лабораторной работы № 1.

2 Напишите программу на языке программирования Python, которая выводит на экран все трехзначные и четырехзначные числа Армстронга. Числом Армстронга называется натуральное число, сумма цифр которого, возведенных в  $n$ -ю степень ( $n$  – количество цифр в числе), равна самому числу:  $153 = 1^3 + 5^3 + 3^3$ .

3 Напишите программу на языке программирования Python, которая запрашивает натуральное число  $N$  и выводит на экран все автоморфные числа, не превосходящие  $N$ . Автоморфным называется натуральное число, если оно равно последним цифрам своего квадрата:  $25^2 = 625$ .

4 Напишите программу на языке программирования Python, которая вычисляет суммы квадратов четных и нечетных чисел от  $m$  до  $n$  включительно и выводит их на экран.  $m$  и  $n$  вводятся с клавиатуры.

5 Напишите программу на языке программирования Python, которая вычисляет сумму  $1 + 2 + 4 + \dots + 2^n$ , где  $n$  вводится с клавиатуры, и выводит ее на экран. Рекомендуется использовать операцию побитового сдвига « $\ll$ ».

6 Напишите программу на языке программирования Python, которая вычисляет произведение  $(1 + 2) \cdot (1 + 2 + 3) \cdot \dots \cdot (1 + 2 + \dots + n)$ , где натуральное число  $n$  вводится с клавиатуры, и выводит его на экран.

7 «Счастливым» лотерейным билетом считается такой, у которого в номере сумма первых трех цифр равна сумме последних трех. Напишите программу на языке программирования Python, которая подсчитывает количество «счастливых» билетов, у которых сумма трех цифр равна 13. Номера билетов шестизначные от 000000 до 999999.

8 Напишите программу на языке программирования Python, которая выводит на экран все простые числа, не превосходящие заданного введенного с клавиатуры натурального числа  $N$ . Простым называется число, которое делится на само себя и 1.

9 Дано натуральное число  $n$ . Напишите программу на языке программирования Python, которая вычисляет значение выражения

$$1 \cdot 2 + 2 \cdot 3 \cdot 4 + \dots + n \cdot (n + 1) \cdot \dots \cdot 2n.$$

10 Напишите программу на языке программирования Python, вычисляющую значение

$$\frac{1}{1 + \frac{1}{3 + \frac{1}{5 + \frac{1}{\dots}}}} \cdot \left( 101 + \frac{1}{103} \right)$$

11 Дано действительное число  $x \neq 0$ . Напишите программу на языке программирования Python, которая вычисляет значение выражения

$$\frac{x}{x^2 + \frac{2}{x^2 + \frac{4}{x^2 + \frac{8}{\dots}}}} \cdot \left( x^2 + \frac{256}{x^2} \right)$$

12 Дано натуральное число  $n$ . Напишите программу на языке программирования Python, которая вычисляет произведение первых  $n$  сомножителей  $\frac{1}{2} \cdot \frac{3}{4} \cdot \frac{5}{6} \cdot \dots$ .

## Лабораторная работа № 8. Оператор цикла while

**Цель работы:** научить студентов применению циклических алгоритмов с предусловием.

### 8.1 Теоретическая часть

В случае, если не требуется организовать повторение некоторого блока кода заданное число раз, а необходимо выполнять его до выполнения (или невыполнения) некоторого условия, применяется так называемый цикл с предусловием `while`. Синтаксис оператора цикла `while`:

```
while expr:
    state1
    ...
    state2
```

Последовательность команд тела цикла будет выполняться до тех пор, пока логическое выражение `expr` равно `True`. В случае, если на входе в цикл `expr` принимает значение `False`, тело цикла не выполняется ни разу. Если значение `expr` всегда принимает значение `True` или эквивалентное, цикл становится бесконечным. Смысл команд `break` и `continue` такой же, как и для цикла `for`.

В языке программирования Python цикл `while` позволяет использовать дополнительную ветвь `else`, которая будет выполнена в случае, если выполнение цикла завершится невыполнением условия `expr` (не командой `break`). Синтаксис оператора в этом случае:

```
while expr:
    state1
    ...
    state2
else:
    additional_state(s)
```

В качестве примера, когда подобная конструкция может быть полезна, – поиск некоторого элемента в списке. В случае, если искомый элемент не найдет выполнение блока, `else` позволит вывести сообщение об этом.

В качестве применения бесконечных циклов `while` можно привести пример контроля ввода. В ряде случаев выполнение программы прерывается сообщением об ошибке, и причин для возникновения подобной ситуации может быть множество, их называют исключениями (`exceptions`). Для перехвата исключений используют инструкции `try – except`:

```

try:
    state1
except except_type:
    state2

```

Когда возникает исключение, интерпретатор прекращает выполнение в блоке try и отыскивает блок except, соответствующий типу возникшего исключения. В случае успеха управление передается первой инструкции в найденном блоке except. После выполнения блока except выполнение программы продолжается с первой инструкции, идущей за блоком try – except. Блоки могут быть вложены друг в друга, однако если интерпретатор достигнет самого верхнего уровня программы и исключение не будет перехвачено, то выполнение программы прервется сообщением об ошибке. Например, следующий блок кода контролирует ввод на предмет ввода вещественного числа:

```

while 1:
    try:
        n=float(input('Введите число'))
        break
    except ValueError:
        print('Вводить нужно число!')

```

Цикл while будет выполняться до тех пор, пока команда n=float(input()) не перестанет вызывать исключения, т. е. ввод числа окажется корректным.

Ознакомиться с типами исключений можно в документации языка Python.

## 8.2 Контрольные вопросы

- 1 В каком случае выполняется блок else оператора while?
- 2 Каким образом в Python можно организовать бесконечный цикл?
- 3 Каким будет результат выполнения программного кода:

```

n=10
while n>1:
    n//=2
    if n==3:
        break
print(n)

```

## 8.3 Задания для самостоятельного выполнения

1 Напишите программу на языке программирования Python, которая выводит на экран разложение натурального числа  $N$ , введенного с клавиатуры, на простые множители.

2 Число, равное сумме всех своих делителей, включая единицу, называется совершенным. Напишите программу на языке программирования Python, которая выводит на экран все совершенные числа в интервале от 2 до  $x$  ( $x$  – целое число, вводится с клавиатуры).

3 С клавиатуры вводятся два натуральных числа  $p$  и  $q$ . Напишите программу на языке программирования Python, которая выводит на экран все делители числа  $q$ , взаимно простые с  $p$ .

4 Напишите программу на языке программирования Python, которая выводит на экран  $N$  первых простых натуральных чисел.

5 С клавиатуры вводятся два натуральных числа  $m$  и  $n$ . Напишите программу на языке программирования Python, которая выводит на экран все меньшие  $n$  натуральных числа, квадрат суммы цифр которых равен  $m$ .

6 Напишите программу на языке программирования Python, которая выводит на экран сумму цифр введенного натурального числа.

7 С клавиатуры вводится подряд последовательность чисел – данные о росте учащихся класса. Сигналом об окончании ввода является ввод числа -1. Выведите на экран средний рост учащихся класса.

8 Напишите программу на языке программирования Python, которая подсчитывает с заданной точностью  $\epsilon$  (действительное число, вводимое с клавиатуры) сумму ряда  $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots$ . Требуемая точность  $\epsilon$  считается достигнутой, если  $|S_{n+1} - S_n| < \epsilon$ .

9 С клавиатуры вводится натуральное число  $N$ . Напишите программу на языке программирования Python, которая выводит на экран все элементы последовательности Фибоначчи, меньшие  $N$ . Последовательность Фибоначчи: 1, 1, 2, 3, 5, 8, 13, 21, 37, ...

10 Натуральное число называется палиндромом, если его запись читается одинаково как слева направо, так и наоборот, например 2552, 1, 252. Найдите все числа, меньшие 100, которые являются палиндромами.

11 Натуральное число называется палиндромом, если его запись читается одинаково как слева направо, так и наоборот, например 2552, 1, 252. Найдите все палиндромы, меньшие 100, которые при возведении в квадрат дают палиндром.

12 Дана последовательность  $\sin x, \sin \sin x, \sin \sin \sin x, \dots$ . Напишите программу на языке программирования Python, которая выводит первый из членов этой последовательности, меньший по модулю, чем введенное число  $\epsilon$ . Значение  $x$  также вводится с клавиатуры.

13 С клавиатуры вводятся два натуральных числа  $n$  и  $m$ . Напишите программу на языке программирования Python, которая находит сумму  $m$  последних цифр числа  $n$ .

## Лабораторная работа № 9. Работа со встроенными функциями

**Цель работы:** ознакомить обучающихся с функциональным подходом в программировании и встроенными библиотеками Python.

### 9.1 Теоретическая часть

Одной из парадигм, поддерживаемой языком программирования Python, является функциональное программирование. Суть этого подхода состоит в определении программы как набора функций и их вычисление в математическом понимании последних (не путать с процедурами в процедурном программировании).

Для определения функций пользователя существует два возможных подхода. В первом случае функция описывается с помощью набора инструкций `def`, во втором применяется так называемая `lambda`-функция.

Синтаксис набора инструкций `def`:

```
def func_name(params, *add_params, **also_params):
    state1
    ...
    state2
    return returned_value(s)
```

Синтаксис `lambda`-функции:

```
lambda arguments: expression
```

Особенностью `lambda`-функций является то, что они определяются в одну строку, могут принимать любое количество аргументов (в том числе и ни одного). Например, функция возведения числа в квадрат может быть описана инструкцией

```
f = lambda x: x * x
```

Когда необходимо использовать функции, определяемые пользователем? В канонах программирования на Python имеется положение, которое утверждает, что любой код, который приходится повторять в программе, следует оформить в виде функции впоследствии вызывать ее в требуемом месте.

В язык программирования Python встроены несколько функций высших порядков (функции, принимающие в качестве аргументов другие функции): `map()`, `filter()`, `reduce()`, `apply()`. Функция `map()` предназначена для пакетного запуска функций, с ее помощью можно передать на вход пользовательской функции набор (кортеж) аргументов и получить набор (кортеж) результатов выполнения. Ознакомиться с действием остальных функций можно, используя систему встроенной справки Python.

Для удобства программиста в стандартную поставку Python входит ряд библиотек, которые условно называются «стандартными». С одной из них, библиотекой `math`, ознакомление осуществлялось ранее. Перечислим некоторые из них:

`cmath` – библиотека математических функций для работы с комплексными числами;

`fractions` – библиотека для работы с дробями и рациональными числами;

random – библиотека для работы со случайными числами;  
 datetime – базовая библиотека для работы с датами и временем;  
 re – библиотека для работы с регулярными выражениями;  
 os – библиотека, реализующая интерфейс с операционной системой;  
 argparse – библиотека для обработки параметров командной строки;  
 urllib – библиотека для обработки URL модулей.

Получить необходимую информацию о методах библиотек можно, используя стандартную систему справки Python.

Особое место в функциональном подходе занимает рекурсия. Рекурсивные программы вызывают сами себя. Рекурсивный алгоритм всегда разбивает задачу на части, которые по своей структуре являются такими же, как и исходная задача, но более простыми. При реализации рекурсивных алгоритмов следует помнить, что в теле рекурсивной функции обязательно должен присутствовать оператор, завершающий рекурсию при выполнении некоторого условия, иначе алгоритм рискует заиклиться и аварийно завершиться, использовав всю выделенную оперативную память.

## ***9.2 Контрольные вопросы***

- 1 В чем состоит суть парадигмы функционального программирования?
- 2 Каким образом в языке программирования Python создается пользовательская функция?
- 3 Что означает термин «функции высшего порядка»?
- 4 Опишите принцип работы функции map().
- 5 Какую стандартную библиотеку Python следует использовать для работы с датой и временем?

## ***9.3 Задания для самостоятельного выполнения***

1 Напишите программу на языке программирования Python, которой на вход подаются тройки чисел  $A$ ,  $B$ ,  $C$ , по которым программа определяет количество корней квадратного уравнения и выводит его на экран. Для определения количества корней квадратного уравнения опишите функцию. Концом ввода троек чисел является ввод слова «end».

2 Напишите программу на языке программирования Python, которая для введенных с клавиатуры троек чисел выводит площадь треугольника с соответствующими сторонами, при этом в случае, если треугольник с указанными сторонами не существует, выводит сообщение «невозможно вычислить площадь по данным параметрам». Вычисление площади и определение существования треугольника оформите в виде функций. Окончанием ввода является ввод слова «end».

3 Напишите программу на языке программирования Python, которая принимает с клавиатуры последовательность чисел и выводит количество чисел, которые являются квадратами некоторых чисел. Окончанием ввода является ввод

слова «end». Опишите функцию, определяющую, является ли поданное ей на вход число полным квадратом или нет.

4 Напишите программу на языке программирования Python, которая вычисляет количество простых чисел в наборе целых чисел, вводимых с клавиатуры. Окончанием ввода последовательности служит ввод слова «end». Оформите в виде функции определение, является ли число простым.

5 Напишите программу на языке программирования Python, которая выводит все числа от  $n$  до 1. Число  $n$  вводится с клавиатуры, разрешается использовать только рекурсию.

6 Напишите программу на языке программирования Python, которая выводит все числа от 1 до  $n$ . Число  $n$  вводится с клавиатуры, разрешается использовать только рекурсию.

7 Напишите программу на языке программирования Python, которая выводит на экран сумму цифр некоторого натурального числа  $N$ . Запрещается использовать циклы, разрешается использовать рекурсию.

8 Напишите программу на языке программирования Python, которая для двух заданных целых неотрицательных чисел  $m$  и  $n$  вычисляет значение функции Аккермана  $A(m, n)$ :

$$A(m, n) = \begin{cases} n + 1, & m = 0, \\ A(m - 1, 1), & m > 0, n = 0, \\ A(m - 1, A(m, n - 1)), & m > 0, n > 0. \end{cases}$$

9 Напишите программу на языке программирования Python, которая определяет, является ли введенное с клавиатуры натуральное число  $N$  степенью двойки. В случае, если является, выводит сообщение «Yes», в противном случае – «No». Запрещено использовать операцию возведения в степень и циклы. Разрешается использовать рекурсию.

10 Напишите программу на языке программирования Python, которая выполняет операции над обыкновенными дробями: сложение и вычитание двух дробей и выводит на экран результаты выполнения действий в виде « $a/b$ ». Используйте стандартную библиотеку `fraction`.

11 Напишите программу на языке программирования Python, которая имитирует игру в кости двух игроков и в конце выводит сообщение «победил XX игрок». Суть игры состоит в бросании подряд 20 раз игральной кости (кубика) и подсчете количества побед каждого из игроков. Для имитации броска игральной кости используйте библиотеку `random`.



## Лабораторная работа № 10. Работа со строками

**Цель работы:** изучить особенности строкового типа данных в Python, ознакомиться с основными методами строкового типа.

### 10.1 Теоретическая часть

Текстовые данные в Python обрабатываются с помощью встроенного класса `str()`, которому соответствует тип данных `str`. Текстовые строки представляют собой неизменяемые последовательности Юникода. Строки в Python можно создать различными способами:

- используя апострофы или двойные кавычки в начале и конце строки, что позволяет использовать внутри строки кавычки другого типа без экранирования;
- тройные кавычки в начале и конце строки, что позволяет создавать многострочные блоки текста;
- из других объектов с помощью методов класса `str()`.

Для того чтобы использовать в строковой переменной непечатаемые символы, используются экранированные последовательности, начинающиеся с символа обратный слэш (`\`): `\n` – перевод строки, `\t` – табуляция и другие.

Если в строковом литерале перед открывающей кавычкой стоит символ `r` (в любом регистре), то механизм экранирования отключается. Такие строки называются «сырыми» (`raw`).

В Python отсутствует как таковой символьный тип, вместо него рассматривается подстрока единичной длины.

Для текстовых строк доступны следующие операции, присущие перечисленным типам:

```

проверка существования значения в строке expr in string;
конкатенация (сложение) строк string1 + string2;
увеличение строки в N раз – повторение строки N раз string * N;
получение символа по номеру i string[i];
получение подстроки (среза) от i-го символа до символа j-1 string[i:j];
получение подстроки от i-го символа до символа j-1 с заданным шагом k
string[i:j:k];
вычисление длины строки len(string);
получение символа с наименьшим кодом min(string);
получение символа с наибольшим кодом max(string);
получение номера первого вхождения подстроки в строку string.index(substr);
получение количества вхождения подстроки в строку
string.count(substr).

```

Больше возможностей для работы со строками дают методы класса `str()`, ознакомиться с которыми можно, используя справочную систему Python.

## **10.2 Контрольные вопросы**

- 1 Какими способами можно задать строковый литерал в Python?
- 2 Каким образом в строковый литерал можно вставить непечатаемые символы?
- 3 Каков результат выполнения кода `"abc"*3` и почему?
- 4 Что такое срез?

## **10.3 Задания для самостоятельного выполнения**

- 1 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке меняет местами первый и последний символы и выводит результат на экран.
- 2 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке удаляет первый символ и выводит результат на экран.
- 3 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке подсчитывает количество цифр (символы от «0» до «9») и выводит результат на экран.
- 4 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке проверяет наличие символов цифр и выводит на экран сообщение об этом.
- 5 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке удаляет последний символ и выводит результат на экран.
- 6 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке удаляет все символы пробела и выводит результат на экран.
- 7 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке удаляет второй и четвертый по счету символы и выводит результат на экран.
- 8 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке удаляет первое слово и выводит результат на экран. Разделителем слов в предложении считайте символ пробела.
- 9 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке подсчитывает количество символов точек и запятых и выводит результат на экран.
- 10 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке подсчитывает количество круглых и квадратных скобок и выводит результат на экран.
- 11 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке добавляет в начало и конец строки символ кавычек.
- 12 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке дописывает справа символы `<_>`, доведя длину строки до 25.

13 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке определяет, совпадает ли количество открывающих круглых скобок с количеством закрывающих.

14 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке после каждой точки ставит еще одну точку и выводит результат на экран.

15 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке меняет порядок символов на обратный.

16 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке меняет первое и последнее слова местами. Разделителем слов считайте символ пробела.

17 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке заменяет каждый символ «\*» числом, соответствующим номеру по порядку вхождения этого символа в строку.

18 Напишите программу на языке программирования Python, которая во введенной с клавиатуры строке меняет порядок следования слов и выводит результат на экран.

19 С клавиатуры вводятся две строки  $s1$  и  $s2$ . Напишите программу на языке программирования Python, которая строит новую строку, состоящую из символов, которые входят как в строку  $s1$ , так и в строку  $s2$ .

20 С клавиатуры вводятся две строки  $s1$  и  $s2$ . Напишите программу на языке программирования Python, которая строит новую строку, состоящую из символов, которые входят в строку  $s1$ , но не входят в строку  $s2$ .

## Список литературы

1 **Гуриков, С. Р.** Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : Форум; ИНФРА-М, 2020. – 343 с.

2 **Гэддис, Т.** Начинаем программировать на Python: пер. с англ. / Т. Гэддис. – 4-е изд. – Санкт-Петербург : БХВ-Петербург, 2019. – 768 с.

3 **Жуков, Р. А.** Язык программирования Python: практикум : учебное пособие / Р. А. Жуков. – Москва : ИНФРА-М, 2020. – 216 с.

4 **Златопольский, Д. М.** Основы программирования на языке Python / Д. М. Златопольский. – Москва : ДМК Пресс, 2017. – 284 с.

5 **Лутц, М.** Изучаем Python: в 2 т. / М. Лутц. – Санкт-Петербург: Диалектика, 2019. – 832 с.

6 **Прохоренок, Н. А.** Python 3. Самое необходимое / Н. А. Прохоренок, В. А. Дронов. – 2-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Петербург, 2018. – 608 с.